An Interop Publication

ConneXions®

The Interoperability Report

March 1994 Volume 8, No. 3

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of ZD Expos, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

ISSN 0894–5926

From the Editor

Portable computers have become almost as common as "desktop" systems. With this trend follows a desire for mobile units to behave just like workstations, particularly with respect to network access. But providing continuous network access to machines that move around is not a simple task. Existing protocols were designed with the assumption of a static network topology where hosts do not change their location over time. Charles Perkins examines this problem further and describes the work of the Mobile IP Working Group of the Internet Engineering Task Force (IETF).

The IETF is also hard at work developing the *Next Generation Internet Protocol* (IPng). We have been promising you a special issue on this topic and are happy to report that most of the material is now in our hands. We plan a publication date of May, which coincides with our NetWorld+Interop 94 conference and exhibition in Las Vegas. By now you should have received the conference brochure for this event. To register, call 1-800-488-2883 or 1-415-578-6900.

In the early days of the Internet, host name to IP address mapping was performed by table lookup. A file, known as `HOSTS.TXT`, was maintained by the Network Information Center (NIC), and host administrators would retrieve the latest version of this file on a regular (or sometimes irregular) basis. This mechanism had several major drawbacks. First, reliance on one central site for file maintenance placed a heavy load on the NIC host. Second, as the network grew, updating the file became an almost minute-by-minute task, and any local copy of `HOSTS.TXT` was always out of date. Third, and perhaps most importantly, the growth of the Internet eventually resulted in a really *huge* `HOSTS.TXT` file and this tied up network and host resources everywhere. Clearly, something had to be done.

A new distributed name–address translation mechanism, known as the *Domain Name System* (DNS), was first described in RFCs in late 1983 by Paul Mockapetris of the University of Southern California's Information Sciences Institute. The system came into production use in 1986–1987, and its deployment has been a major contributing factor to the success of the Internet.

But simply having access to DNS software does not guarantee smooth operation of the Internet. Participants in the DNS system must follow certain guidelines for configuration and operation. Bryan Beecher from the University of Michigan has collected *The Ten Commandments of Domain Name Service* which we present starting on page 21.

Mobile IP as seen by the IETF

by Charles E. Perkins, IBM, T. J. Watson Research Center

Abstract

Due to advances in wireless communication technology there is a growing demand for providing continuous network access to the users of portable computers, regardless of their location. Existing network protocols cannot meet this requirement since they were designed with the assumption of a static network topology where hosts do not change their location over time. The *Internet Engineering Task Force* (IETF) has developed a model which fits naturally into the usual framework for routing over *Internet Protocol* (IP) networks. The requirements which shaped the evolution of the IETF model are outlined. The entities characterized by the model are defined, and their interactions described. A suggested way in which these interactions occur within the existing infrastructure is mentioned, and proposed extensions are briefly outlined. A proposed interface to Layer 2 is also shown.

Introduction

The Internet Engineering Task Force is an internationally recognized group of network engineering experts which meets three times a year. There are about 70 "working groups," which convene at the IETF meeting sites, but which also conduct a significant amount of their business by mailing lists. Each group focuses on a different area of investigation; there are numerous groups looking at various routing protocols, ATM operations, Domain Name System extensions, and many others. In particular, the "Mobile-IP" working group has been organized to investigate the protocol requirements and techniques for correct operation of mobile computers from the Layer 3 perspective. It is by no means sufficient for a Working Group to achieve consensus at one of the thrice annual meetings, although that is highly desirable. Consensus requires also that all the mailing list participants from around the world have a chance to review any proposed agreements, and make technical comments.

As time has passed, the Mobile-IP working group of the IETF (which will be called the Mobile-IP WG) has studied numerous proposals for enabling mobile networking. Each proposal has had advantages and disadvantages. Each proposal has effectively advanced the "state of the art," so that by now we feel that we (collectively) have a pretty good handle on the overall problem area. My intention in this article is to discuss the goals, problems, and solutions for mobile networking. I will explain the Mobile-IP WG model for mobility, and hopefully put forth a convincing case that this model provides a fairly complete solution which naturally fits into Layer 3 (the network layer) of the commonly accepted protocol stack. We have framed the problem essentially as a packet forwarding problem—in other words, we have been trying to solve the problems presented by mobile computers, by finding ways to forward packets to wherever they may currently be located.

Goals

Users of mobile computers will basically desire to use their mobile units as easily as today's office dwellers use desktop computers. We all know the benefits of having large networks of computer resources. People with mobile computers will naturally expect to have the same resources available to them. This will include the current methods for interpersonal communications. Ideally, and practically, this extra facility for mobility must be available without requiring user intervention. Clearly, we want to engineer our solutions so that network efficiency is not affected; this means, for instance, that a good solution will not use features that cause router performance to deteriorate markedly.

Just as clearly, we cannot institute a protocol which requires undue host processing. The most important requirement is that the solution must provide the same network connectivity to network resources as is available today. That implies compatibility with existing computer equipment, and application transparency. The solution cannot require expensive equipment, compared to the computer itself. At the Mobile-IP WG, we decided on three "hard" requirements:

- Interoperability with present computer equipment and applications
- Continuous access across multiple networks
- Security as good as with existing networks

The first hard requirement is obvious. We can't expect people to use a solution that does not let them access their existing resources. Meeting the second criterion allows people to use mobile computers in about the same way as they use stationary computers, within obvious physical limits. And, while the Mobile-IP WG recognizes that mobile computers seem to create additional security problems compared to stationary computers, nevertheless we have, tentatively, decided to avoid making security the main focus of this work. We intend to analyze fully the security exposures of the protocol, and to make it easy to insert any reasonable security mechanism when it becomes available. There is another IETF working group charged with the job of specifying a good security procedure for IP, and we will await the results of their efforts. There are people who are involved with both groups.

Problems

Historically, a "network" corresponded nicely to a long piece of wire. Maybe the network wasn't really a piece of wire, but the equipment manufacturers were nice enough to make it seem so. One way of expanding a network to accommodate more computers is to add more equipment, and more wires connected together with things like repeaters and bridges. However, eventually this strategy runs out of steam, and different networks must be connected together. The different networks occupy different sequences of addresses within the network layer addressing scheme, and packets between networks are handled by network layer routers. Such has been the model used very successfully by the Internet Protocol. However, it is not immediately clear what happens when no direct relationship can be made between a Mobile Host's network address and a piece of wire connected to a router.

Since hosts were thought of as being hooked up to a network, it was natural enough to encode network identification information within the network layer address. This network layer address now serves at least two distinct roles. One, it serves as an "endpoint identifier"—a way to specify what entity resides at the endpoint of a communication path between two hosts. The other use for the network address is as a roadmap for locating the computer within the Internet. This roadmap is followed by isolating out the network identification part of the address, and subsequently forwarding the packet to whatever agent is currently routing packets to hosts on that network. This is usually done using a simple hop-by-hop algorithm whereby each intermediate router keeps track of what the next hop is along the way, for each particular destination network.

continued on next page

Mobile IP as seen by the IETF *(continued)*

Then, once the particular network (wire, perhaps?) is reached, the remaining network layer address bits specify which host on the network should receive the packet. Once the packet reaches the appropriate network, it is assumed that the particular host involved will receive the packet. It is this last assumption which breaks the worst in the realm of wireless networks. Even given a consistent model of a wireless network which has a router advertising reachability to that network, it is mostly not true that the packet will get to the particular destination without the active and special intervention of the router. When computers are mobile, the network address no longer specifies the location.

Mobile computers introduce another new problem that has never been dealt with by IP and most other network layer protocols—that of tracking the movements of the computers. It is not a surprise, then, that IP doesn't have any built in facilities for handling this, unless one counts the ability to change routes dynamically. Issuing location updates in response to movements of the mobile computers will be a central feature of any acceptable mobile networking protocol. Controlling the dissemination of this data turns out to be of central importance, and the matter has yet to be fully resolved. Clearly, one would like to make sure that current location information is available wherever it is needed, but nowhere else. And, the level of need is not a simple binary decision. The wider dissemination of location data is advantageous in allowing better routing, but disadvantageous when considering how to eliminate stale information.

Of course, one of the prime requirements for any workable system is compatibility with existing systems. This translates directly to the requirement that mobile computers using any protocol we specify must be able to maintain network connections with existing hosts, using the routing services of existing routers. We can add equipment (e.g., mobile computers, wireless transceivers) but we can't design a system that doesn't work with today's computers. This same requirement applies to application software, which is the only reason most people have computers. Mobile computers using a new protocol cannot expect to recompile or re-link existing application software, nor run only software which is newly designed to enable mobility. The best we can expect is that necessary protocol changes will be installed into the protocol service itself, and only there, on the additional equipment.

One consequence of application transparency, as mentioned above, is that the network address of the mobile computer cannot change when the computer moves. Otherwise, movement of the computer might require rebooting or restarting applications, which seems to be unacceptable from the perspective of user convenience.

It may be quite important for the new protocol and procedures to allow the determination of optimal paths to the mobile computer, even as the user moves from place to place. This ability to dynamically determine optimal routes in the face of changing information is a very hard problem in general, but fortunately for us there is a big restriction we can place on the general problem. Namely, we can assume that the mobile computers get access to the rest of the network through fixed attachment points. The attachments points are only a single hop away from the mobile computers, so finding the mobile computer is about the same as finding its current attachment point. Nevertheless, choosing a strategy to allow the optimization of routing paths to mobile computers turns what might be a pretty simple problem into a matter requiring great consideration.

Last, but not least, it must be emphasized that security in mobile networking is a matter that must be addressed, especially for wireless mobile networking. This has implications for every aspect of the design, and in some cases has been the determining factor for the relevant design decisions.

TCP/IP

IP stands for *Internet Protocol,* and is a widely used, connectionless or "best-effort" Layer 3 protocol [1, 20]. The initial development of IP was funded by DARPA, and it gained widespread popularity with the development and distribution of BSD UNIX from the University of California at Berkeley. IP "internetworks" separate networks together, to form a cohesive routing infrastructure in which (usually) any host computer with an IP address can exchange packets with any other host computer possessing an IP address. The Internet, which runs IP, has grown to the point of offering worldwide connections to over two million computers, a feature no other computer network can offer. Moreover, the Internet has continued to double in size for several years now, and does not seem to show much sign of slowing down.

Routers are responsible for delivering packets from one network to another within this infrastructure. However, IP does not *guarantee* delivery of a packet to its eventual destination. It merely acts upon the best information available to deliver the packet hop by hop to the destination. If, at some stage, there is inaccurate or stale routing information, packets will possibly be lost. If a particular destination becomes momentarily uncommunicative, IP will not take any additional measures to retry the transmission later. In fact, each packet may be delivered along different routes, which then may require different times to accomplish the end-to-end delivery. This may result in packets being delivered out of order. Correcting these errors and possibly others is the job of higher-level protocols, in particular the *Transmission Control Protocol* (TCP) [2].

TCP sequences packets and presents the appearance to its applications of a reliable data stream. TCP accomplishes this by (among other things) retransmitting packets when they appear to have gotten lost. In addition to this, TCP takes certain measures to avoid network congestion as well as host application overruns, so that an application using TCP can count on several flow control features as well as the basic reliable data service. TCP even performs source route reversal on behalf of its applications.

We have chosen to frame the requirement of enabling computers to move freely, as a routing problem. That is, when a computer moves from one place to another, we characterize the design problem as one of ensuring that packets can be forwarded to the current location of the computer. Thus, we choose to make computers mobile by discovering what needs to be done to the Internet Protocol. This will have the effect of making all IP applications transparently mobile, without requiring any changes to them. The applications most likely will not be able to tell when the computer has moved.

Definitions

In this section, we will define terms, and describe a basic model. We will describe the basic relationship between the entities populating this model system.

Host: Any computer, not considered to be performing routing or bridging functions.

Mobile Host: A Host which moves from place to place, invalidating historical Internet design assumptions about static placement of computers.

continued on next page

Mobile IP as seen by the IETF *(continued)*

Correspondent Host: A Host communicating with another Host; particularly in most relevant discussions, communicating with a Mobile Host.

Home Address: An address used to identify a Mobile Host no matter where it may currently be located (cf.: discussions about "endpoints").

Foreign Address: An address used to locate a Mobile Host at some particular instant of time.

Foreign Agent: A specialized forwarding agent which offers a Foreign Address, and maintains and performs a mapping between that address and the Home Address of a Mobile Host in its care.

Home Agent: An agent that redirects or tunnels packets from a Home Network to the Foreign Address of a Mobile Host.

Home Network: The (logical) network on which a Mobile Host's Home Address resides.

Ad-Hoc Networking: Networking between Mobile Hosts in the absence of any other agents.

Triangle Routing: A situation in which a Correspondent Host's packets to a Mobile Host follow a path which is longer than the optimal path because the packets must be forwarded to the Mobile Host via a Home Agent.

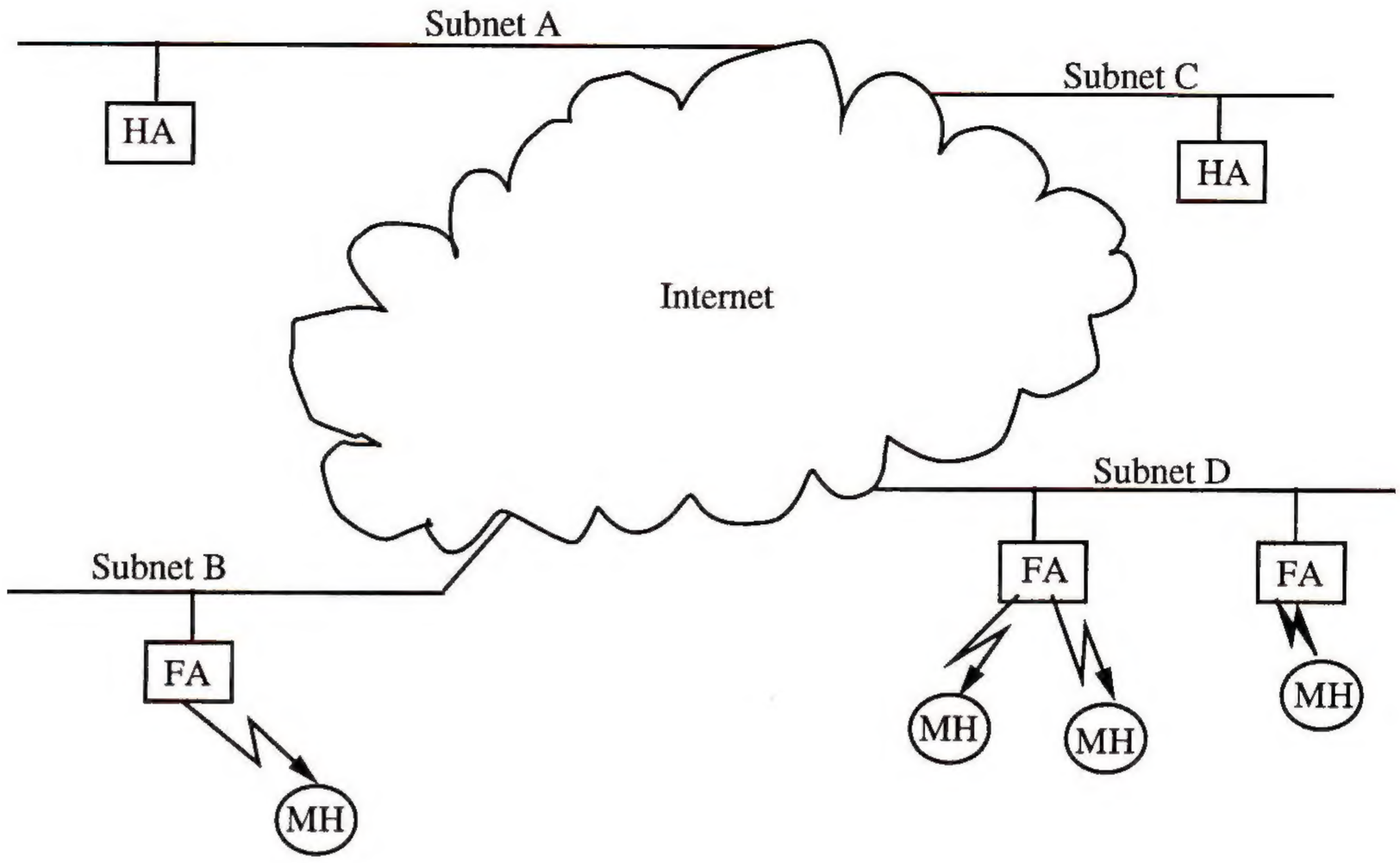
Weak Security: The same level of security provided by today's Internet. A weakly secure system does not protect against snoopers popu-
lating the normal path along which a packet would traverse.

General description of model entities

Our system involves participation of three types of entities, viz., *Mobile Host, Foreign Agent* and *Home Agent.* The networking architecture we assume is that of a set of Foreign Agents connected through a wired backbone. A Foreign Agent supports at least one interface which is being made available to Mobile Hosts, and makes available a kind of forwarding address (the Foreign Address) to any Mobile Host accepting its services.

Within one campus or administrative domain there could be multiple Mobile Subnets. Each Mobile Subnet can be handled by a separate Home Agent, or the same Home Agent can handle several otherwise unrelated Mobile Subnets. Unlike other routers, a Home Agent is not required to have an interface corresponding to the Mobile Subnet it serves. The association between each Mobile Host and its current Foreign Address is kept at least by its Home Agent.

A Mobile Host retains its Home Address regardless of its current Foreign Address. It can start sessions with other hosts (both mobile and stationary) and obtain new Foreign Addresses without disrupting any active sessions. The movement of a Mobile Host is completely transparent to the running applications, except possibly for a momentary pause which may occur during the transition. A Mobile Host has only one Foreign Address at any given time. Even if two different Foreign Agents are serving the same physical area, a Mobile Host accepts service from only one of them. A Foreign Agent can offer service to multiple Mobile Hosts. There isn't any relationship in general between the Foreign Address of a Foreign Agent and the Home Network which the Mobile Host belongs to, and similarly there is no relationship between the addresses of the Foreign Agents and the Home Agents. The relationship between the various mobile networking entities is illustrated in Figure 1.



(HA=Home Agent, FA=Foreign Agent, MH=Mobile Host)

Figure 1: Mobile Networking Entities in the Internet

We use the term *Correspondent Host* to refer to the host communicating with an Mobile Host. In the following discussion, a stationary correspondent host is also referred to as *Stationary Host.*

General operation of the IETF model

As a result of the way we have framed the problem, solutions occur more naturally within the visible design space. We basically propose that the movement of Mobile Hosts can be enabled by solving a simply stated routing problem. Namely, we can achieve our goals by finding a way to route packets between the Home Agent and the Foreign Agent. Packets destined for a Mobile Host will, unless special measures are taken, be forwarded to the Home Agent by normal IP forwarding methods, because the Home Agent advertises connectivity to the Home Network specified by the address of the Mobile Host. Once packets get to the correct Foreign Address (i.e., the Foreign Agent), they will be delivered correctly by the natural action of the Foreign Agent. And, we can get packets from the Home Agent to the Foreign Agent by the simple expedient of encapsulating them and inserting the Foreign Address as the destination IP address of the encapsulated packet.

A location update function is also needed, to allow the Home Agent to know where all of its Mobile Hosts are. And, lastly, the Mobile Host needs to complete a transaction with the Foreign Agent before the Foreign Agent will agree to make its advertised services available to the Mobile Host. But these functions, important as they are, should not obscure the simplicity of the model and the approach of providing a wide-area solution based on inter-network routing. Internetworking is IP's strength, and we expect to use that strength to good advantage by providing this natural model of operation for Mobile Hosts. Even with no further protocol operation it should be clear that this basic model, using encapsulation, can provide the basic routing services needed.

The next section discusses some previous proposals for Mobile Host support.

continued on next page

Mobile IP as seen by the IETF *(continued)*

IBM Loose Source Routing

This proposal, by Rekhter and Perkins [17, 21], took advantage of the existing IP option which allows hosts to specify intermediate forwarding nodes for return traffic. If each Mobile Host specifies that its current Foreign Agent (access point) is in the source route, then its Correspondent Hosts should be able to return packets to it by the natural route reversal action specified as part of the *Loose Source Route* option handling. Ideally, this would allow a fairly simple solution within the existing infrastructure, and in addition the routing paths taken would usually be close to optimal. Unfortunately, not many hosts implement Loose Source Routing correctly even though it has been part of the IP specification for a long time. Worse yet, hosts correctly processing Loose Source Route options are susceptible to a fairly serious breach of security. Moreover, existing routers are optimized to handle packets quickly if they don't have any IP options. That means, the existing routers will perform badly if they have to forward a lot of packets with IP options; there have already been instances where significant amounts of source routed traffic have caused major difficulties within the Internet. Thus, any solution involving IP options is unlikely to be popular with the administrators involved with the maintenance of high-traffic backbone routers. Lastly, in the particular case of the Loose Source Route option, no known UDP applications do the route reversal correctly, so that popular services such as NSF and DNS cannot make use of the desirable route optimizations offered by the Loose Source Route proposal.

However, aside from the particular means of route optimization and delivery of packets from the Home Agent to the current Foreign Agent serving the Mobile Host, the overall appearance of the model used with the Loose Source Route proposal resembles the current proposal.

Sony VIP

Fumio Teraoka at Sony developed an approach [8, 11] which modeled the Mobile Host as possessing a Physical Address and a Virtual Address. The Sony approach relies on a generalized "Virtual Network," which when implemented for IP subdivides Layer 3 protocol processing into two parts, one for handling each of the addresses associated with the Mobile Host. This handling can be specified either by a new IP option, or by a new protocol number which triggers the sublayer processing within IP. Either way, the IP implementation is known as *VIP* (Virtual IP). The Physical Address of the Mobile Host is obtained by whatever means when the Mobile Host moves to a new location, and the Virtual Address corresponds to a network (real or virtual) which is served by a specialized router. The network served by this router is called the *Home Network.*

To handle the case of existing hosts, the VIP approach suggests that these specialized routers snoop packets while forwarding them. If the packets emanate from a Mobile Host, and thus contain both a Physical Address and a Virtual Address, Sony routers can glean this information for possible future use. The information gleaned is put into an *Address Mapping Table* (AMT), whose entries have timeouts and must be managed by deletion or update according to the timestamps of new packets passing by. The next time such a router forwards a packet destined for the Virtual Address of the Mobile Host, the Sony router can modify the IP header so that it shows the Physical Address of the Mobile Host as its new destination. Thus, existing hosts will send out packets destined for the Home Network of the Mobile Host, but that will quickly be corrected so that the packets are destined for the current location of the Mobile Host.

Carlberg's Host Route

Since existing routers can support host routes (i.e., routes to a specific host that do not imply reachability to other hosts on the same "network"), one approach might be to have the Foreign Agents be routers. Then they could advertise reachability to just those Mobile Hosts which were currently located within their range. This is the basic approach advocated by Ken Carlberg [19] in an earlier paper, although in the context of ISO/GOSIP Mobile End Systems.

While this approach has the advantage of simplicity, ease of understanding, and lack of reliance on new protocols, it seems likely to fall prey to problems of scale. In an organization with hundreds of mobile hosts, and dozens of routers, it is easy to imagine that the constantly required updates flowing through the organizational networks would start to cause difficulties. In the larger Internet, it seems that supporting arbitrary mobility is not feasible.

Columbia (JI) MSSs

One popular approach was specified and implemented at Columbia University as part of the Ph.D. requirements of John (JI) Ioannidis. JI's approach [13, 14] modeled mobility between networks of a campus and solved the problems by the use of *Mobile Support Systems* MSSs (also called Mobile Support Routers (MSRs)). The MSSs share information about the current whereabouts of Mobile Hosts, and when a Mobile Host moves to a new network, the new MSS and the old MSS coordinate the forwarding of packets to the Mobile Host. The Mobile Host always knows the address of its MSS, so this is feasible. The MSSs conspire to provide the appearance of a virtual mobile subnet spread over a campus-sized number of real subnets, and use a *Mobile Internetworking Control Protocol* (MICP) for communicating among themselves the location information needed for the Mobile Hosts.

JI's approach specified the use of an encapsulation protocol, IPIP, for delivering packets from one MSS to another. The MSSs cooperatively act as a distributed router for the virtual subnet, and are known to the routers for the other subnets to have reachability to that virtual subnet. JI also specified a beaconing protocol so that the Mobile Hosts could discover which MSS was serving their region.

For mobility outside the campus, JI added a "popup" feature to this basic protocol setup. The Mobile Host would acquire a local address outside the campus, and report that address to one of its home MSSs as its new location. Thus the Mobile Host would effectively act as its own MSS. The Home MSS tunnels packets to the popup, and advertises to the other campus MSSs that it is currently serving the popup Mobile Host.

Matsushita

Another approach, by Tatsuya Ohnishi, Hiromi Wada, and Brian Marsh of Matsushita [12], specifies the use of *Packet Forwarding Servers* (PFSs). Packet Forwarding Servers operate somewhat like Sony routers by intercepting packets destined for Mobile Hosts and forwarding them to the current location of the Mobile Host. A Mobile Host has a Home Address, and when it reconnects, it gets a new temporary IP address. Modified stationary hosts cache the binding between the Mobile Host's two addresses, but unmodified hosts just transmit the packet to the Mobile Host's Home Address. If there aren't any PFSs which know the current binding, the packet will eventually arrive at the home PFS, which will be able to encapsulate the packet (using the *Internet Packet Transmission Protocol,* IPTP) and deliver it to the Mobile Host.

continued on next page

Mobile IP as seen by the IETF *(continued)*

When a Mobile Host gets a temporary address on a new network, it tries to find a local PFS by using a ping operation. If a PFS replies, its existence will be made known to the Home PFS. Then other Stationary Hosts on that network will be notified by the local PFS if they send packets to the temporary address of the Mobile Host. The local PFS will be notified by the Home PFS when the Mobile Host moves to a new network.

IBM Re-addressing

This proposal (by Perkins and Rekhter) [22] is an attempt to abstract and generalize previous approaches. Except for Carlberg's "host route" idea, all the other proposals assume the existence of a special "Mobile Network," and a specialized router or cooperating set of routers advertising reachability to that network. The techniques of encapsulation or source routing can be both used to deliver packets from the Mobile Network to wherever the particular Mobile Host is currently located. In the description of this proposal, Mobile Hosts are the clients of Re-addressing Servers, which can readdress packets destined for one of their clients by either stripping the encapsulation or managing the source routes according to IP specifications. In addition to remembering which Mobile Hosts are its clients, a Re-addressing Server also may maintain a cache of useful information about the Correspondent Hosts with which its clients may wish to communicate. When the Correspondent Host is itself a Mobile Host, this cache would contain the current location information about that Correspondent Host. In this way, a Re-addressing Server can help its clients obtain optimal routing whenever possible. Correspondent Hosts which can perform their own re-addressing can also maintain similar caches. These hosts are characterized as being served internally by their own Re-addressing servers. Thus optimal routing can be achieved whenever a pair of Re-addressing Servers are communicating with each other. One advantage of this approach is that multi-level mobility can be achieved by creating nested Re-addressing Servers.

CMU MHRP

Dave Johnson from CMU proposed a variant using Loose Source Routing to effect the management of packet delivery to Mobile Hosts. This was subsequently modified to use a new encapsulation technique which allowed lazy updates to the location information cached at the previous connection points of the Mobile Hosts. The new proposal [24] was called the *Mobile Host Routing Protocol* (MHRP). The Mobile Hosts are modeled, again, as having addresses on a Mobile Network upon which resides a Location Server which is capable of delivering packets to the Base Stations which are at the current locations of the Mobile Hosts. The Base Stations themselves maintain Location Caches for traveling Correspondent Hosts and previous clients (i.e, Mobile Hosts which have moved). Any agent which wishes to deliver a packet to a Mobile Host, and knows about the current location of that Mobile Host can encapsulate the packet so that the current location becomes the destination. In this way, optimal routing can be achieved. If the Location Cache is stale, and thus a packet is delivered to the wrong current location of a Mobile Host, the agent at the end of the tunnel changes the destination to reflect the more current information and adds its own address as part of the data encapsulated by the newly modified IP packet header. When the packet finally arrives at the correct current location of the Mobile Host, all stale cache entries in the intermediate Location Caches can be updated to contain the current information. Of course, there will rarely be any times when more than one intermediate Location Cache has a stale route entry.

One feature of MHRP is its attempt to specify a minimal encapsulation protocol for the needed tunnelling. In many cases, the encapsulation only requires 8 additional bytes per packet destined for a Mobile Host, in contrast to earlier approaches which used 20 or more bytes per encapsulated packets. In addition, MHRP tried to attend to the necessary details to allow a Mobile Host to interact correctly with other hosts even when directly attached to its Home Network. In this way, MHRP could be used to effect wired mobility, if for instance a Mobile Host was to be moved from one Ethernet network to another. This could, conceivably, be done without losing any active network sessions, since the physical operation still makes sense. Such wired mobility might not make sense on other kinds of physical networks.

Myles/Perkins MIP

Building on ideas found in IBM's proposals, MHRP, and Sony's VIP, Myles and Perkins [23] tried to build an improved version of MHRP using a new IP option, the MIP option. The basic model of a Mobile Network served by a Home Agent, and *Internet Access Points* (IAPs) offering service to Mobile Hosts on arbitrary networks, is preserved. But the Location Caches at the IAPs have timeouts associated with them and can be refreshed by MIP packets, which now have timestamps in them. MIP explicitly offered equivalent operation whether the IP option, or MIP encapsulation, was selected, because we wanted the protocol selection to depend mainly on the specific mechanisms offered, not on the variety of readdressing used to effect those mechanisms. Although technically there may be some slight advantages to using IP options, the disadvantages detailed above for Loose Source Routing made it clear that encapsulation had fewer objections from a practical standpoint.

MIP also specified a method for intermediate routers to participate in optimization of routes between unimproved existing hosts and Mobile Hosts. This operates similarly to the Sony router approach. Much attention was spent trying to improve the reliability of protocol operation in the face of various network or MIP entity failures.

SMIP (CDPD-like)

Simple MIP (*Mobile Internet Protocol*), or SMIP, was released by Penners and Rekhter [26] as a simplified approach to mobility without worrying about achieving optimal routing. The concern was that previous attempts to optimize the routing of packets between Mobile Hosts and their Correspondent Hosts left unsolved certain problems of security. For one thing, it was not clear how the integrity of Location Caches would be maintained in the face of a determined opponent wishing to usurp communications to a Mobile Host. For another thing, the various kinds of management information flowing through the network, especially the location updates, could be used by an unfriendly snooper to discover private information about the current location of a Mobile Host. SMIP also was based on the premise that no one could really quantify the extent to which optimal routing was needed, or indeed if it were even likely to be provided by the other proposals. Most of the other existing proposals certainly work best in the hypothetical future when the existing infrastructure has finally been upgraded to fully support mobility.

Another distinguishing characteristic of SMIP was its relatively new approach to cell discovery (left unspecified in MIP and MHRP and others). The SMIP approach mimics the CDPD specification in its overall aspect, and for the first time proposed that the Visiting Register (providing the temporary or "Care-of" address for the Mobile Host as it moves) intercede for the Mobile Host in negotiating the transmission of the location update to the Home Register which plays the part of the Home Agent in SMIP.

continued on next page

Mobile IP as seen by the IETF *(continued)*

Either the Visiting Register or the Home Register in SMIP can refuse the new connection if necessary. The same method of approving new connections by multiple registration steps is found in CDPD.

The recognition by Penners and Rekhter of the need to consider the security aspects of transmitting location updates has had a large effect on the recent thinking of the IETF. In addition, the multi-step registration model has been adopted, presumably for the additional control it makes available as well as to leave open the option for possible convergence or simplifying multi-targeted implementations for both IETF and CDPD.

Current IETF direction

Having developed a generally agreed-upon model of the system, the next task is to make certain design decisions, going from the abstract to the specific, until a concrete protocol results. This section will attempt to list and justify various design decisions that have been made up until now by the IETF Working Group. Some of the decisions have been made more firmly than others, and where appropriate the degree of consensus will be indicated.

Encapsulation

In order to deliver a packet from the Home Agent to the current location of the Mobile Host (i.e., its Foreign Agent), the packet has to be "re-addressed." Otherwise, as soon as any intermediate router tried to forward the packet, the packet would be delivered back to the Home Agent just as it was the first time. This readdressing can occur by using a source-routing technique, or by actually changing the destination of the packet to be that of the Foreign Agent. Although architecturally elegant, source routing techniques have been found through hard experience to have certain grave drawbacks in practice, given the structure of our current Internet. Among the several problems, we have found that Loose Source Routing:

- Slows down intermediate routers
- Exacerbates existing security problems
- Is implemented incorrectly by TCP in most computers, and
- Is not handled correctly by any known UDP applications

Since it is unlikely that existing computers will repair their implementation of Loose Source Routing any time soon, we have to accept the current situation as a characteristic of the existing infrastructure regardless of any protocol specification. The additional security exposure is presented by using Loose Source Routing because the common schemes propose to include the Foreign Agent's address as a component of the source route. Without additional precautions, this would allow any computer within the Internet to pose as the Foreign Agent for a Mobile Host, and thus be able to inspect a stream of data between that Mobile Host and any Correspondent Host. The interloper would thus be able to pose as any desired Correspondent Host.

Thus, encapsulation has been selected as the preferred method of readdressing. Encapsulation means that data from the original IP header is prepended to the IP data field, and the original header is then modified to contain different destination and protocol fields. Other fields may have to be changed too, depending upon the exact form of encapsulation chosen by the IETF. Nothing in the protocol depends directly on the style of encapsulation which will be chosen, but it is likely that one style will be required to be supported by all mobile entities involved. Other styles of encapsulation might be supported after optional negotiation steps.

Solicitation

When a Mobile Host cannot detect any advertisement of available services from a Foreign Agent (see the next subsection), it may decide to solicit service. This might happen, for instance, if a Mobile Host was moving from one wired network to another. In that case, there might be a Foreign Agent making a Foreign Address available on the wired network, but which was not wasting any wired bandwidth with periodic beacons. The device driver for the wired Mobile Host might then have to detect the presence of carrier or some other condition on the wired network. When the condition was satisfied, then the Mobile Host would perform the necessary solicitation for service.

The following fields are good candidates for inclusion in the solicitation packet transmitted by a Mobile Host:

- Type, code, checksum
- Mobile Host's IP address
- Mobile Host's MAC address

Note that the last two fields are found in the packet headers for layers 3 and 2 respectively, and so do not necessarily have to be included in the data part of the packet.

The solicitation would have to be either broadcast or multicast, because the Mobile Host would not necessarily have, a priori, any indication of the local Foreign Agent's MAC address. Similar considerations might apply to wireless connections also. That is, a Mobile Host might decide to send out a solicitation for service whenever the current Foreign Agent seemed unavailable. Solicitations are useful whenever there is no periodic beacon (service advertisement), and perhaps even when the periodic beacon has been delayed or omitted for whatever reason.

Advertisement

As indicated in the last section, a Mobile Host expects to receive service advertisements from a local Foreign Agent. The exact format of this advertisement remains incomplete, but certain fields are very good candidates for inclusion, for instance:

- Type, code, checksum
- Foreign Address
- Foreign Agent incarnation number
- Advertisement interval

Also, just as a natural part of the transmission of the advertisement packet, all potential clients will discover the base station's MAC address and its IP address. This IP address may be distinct from the Foreign Address, however, because the base station may be distinct from the Foreign Agent; for example, there might be a wired network with a single Foreign Agent and multiple base stations. The exact method for transmitting packets between a Foreign Agent and the base stations in this situation is not going to be specified by the IETF Working Group, but we have to be aware of this possibility and make sure the protocol allows for correct operation in this case.

The incarnation number is included for use by the Mobile Hosts when a Foreign Agent crashes. Whenever the Foreign Agent crashes, it will have to increment its incarnation number. When any Mobile Hosts, which remain clients of the Foreign Agent, discover that the incarnation number has changed, they will know that the Foreign Agent may have forgotten about them.

continued on next page

Mobile IP as seen by the IETF *(continued)*

That Foreign Agent may have to be reminded to provide service for the Mobile Hosts. This would presumably look much like the process of entering the service area of a new Foreign Agent.

If the Foreign Agent sends out a service advertisement in response to a solicitation made by a Mobile Host, then the response can be sent out as a unicast message instead of a broadcast or multicast (beacon) message.

Mobile Host <—> Foreign Agent Registration

The Registration protocol will probably consist of 4 messages. First, is a registration message between the Mobile Host and the Foreign Agent, covered in this subsection. In the next subsection the registration between the Foreign Agent and the Home Agent on behalf of the Mobile Host will be discussed. Finally, there are two acknowledgment messages corresponding to these two registration messages. It is possible to imagine that the same packet type might perform both positive acknowledgment functions, as well as negative acknowledgments.

Once the Mobile Host discovers that it can get service from a Foreign Agent, it must register with that Foreign Agent. Likely candidates for inclusion in this registration packet are as follows:

- Type, code, checksum
- Home Agent's address
- Sequence number
- Previous Foreign Address
- Mobile Host authenticator to Home Agent
- Mobile Host authenticator to Foreign Agent
- Mobile Host's IP address
- Mobile Host's MAC address

Notice again that the Mobile Host's Layer 2 and 3 addresses may be found in the registration packet's headers.

The first authentication field is present so that the Home Agent can be somewhat assured, as described in the next subsection, that the information about the Mobile Host is indeed coming from that Mobile Host and not an impostor. The Mobile Host also passes a different authentication to the Foreign Agent, because the Foreign Agent may receive update information regarding the Mobile Host in the future, and it would be valuable for the Foreign Agent to discover whether the update information was authentic or not. Note that these are not particularly strong measures for ensuring authenticity, and when more security is required different mechanisms will have to be built. We expect that these different mechanisms will use registration packets with a different (type, code) pair, even if the rest of the registration protocol is the same.

The sequence number field plays the same role as a timestamp, so that delayed and out-of-order packets do not cause any interruption of service. Otherwise, a stale packet from a nearly-recent cell switch might cause confusion upon its eventual arrival at a Home Agent. This sequence number is the one included in all management packets related to this particular location information for the Mobile Host.

Foreign Agent <—> Home Agent Registration

When the Foreign Agent is satisfied that it has been contacted by a Mobile Host which is a good citizen, then the Foreign Agent can initiate the next registration phase, by registering its new client with the client's Home Agent. The registration packet will be likely to contain the following information:

- Type, code, checksum
- Foreign Address
- Sequence number
- Mobile Host authenticator to Home Agent

There may also be included the address of the base station if that IP address is ever distinct from the Foreign Address; this issue has not been resolved. The sequence number is the same one transmitted by the Mobile Host to the Foreign Agent, and similarly for the authenticator. Notice that the packet can usually be delivered to the Home Agent simply by sending it out with the destination address equal to the IP address of the Mobile Host. Such packets arrive at the Home Agent by the normal operation of the existing Internet, since the Home Agent is well-known as the router for the Home Network. This would relieve the Mobile Host of the administrative requirement to be configured with the address of its Home Agent.

Registration Acknowledgments

The Home Agent will send a packet back to the Foreign Agent either accepting or rejecting the Mobile Host's request. A positive acknowledgment will be likely to contain the following information:

- Type, code, checksum
- Mobile Host's IP address
- Sequence number
- Foreign Address
- Cache timeout

The cache timeout specifies how long the Mobile Host can trust the Home Agent to keep track of the Mobile Host's whereabouts. If it is not infinite or zero, then the Mobile Host should attempt to re-register as often as is indicated by the timeout. This feature was included to avoid problems caused by crashing Home Agents. If the Home Agent crashes, and the Mobile Host never re-registers, then it would be possible for the Mobile Host to be unaware that the Home Agent is dropping all packets which otherwise would have been delivered to the Mobile Host.

The Foreign Agent also needs to deliver an acknowledgment to the Mobile Host. This acknowledgment contains the following:

- Type, code, checksum
- Mobile Host's IP address
- Sequence number
- Cookie value
- Cache timeout
- Cell expiration

The cache timeout is the one received from the Home Agent, and the cell expiration value specifies how often the Mobile Host should re-register with the Foreign Agent. The cookie value can be used by the Mobile Host to authenticate future control messages to this Foreign Agent.

continued on next page

Mobile IP as seen by the IETF *(continued)*

For instance, when the Mobile Host moves to a new Foreign Agent, the Mobile Host may wish to send a control message to its previous Foreign Agent, to notify the previous Agent of the change. Figure 2 illustrates a typical registration sequence.

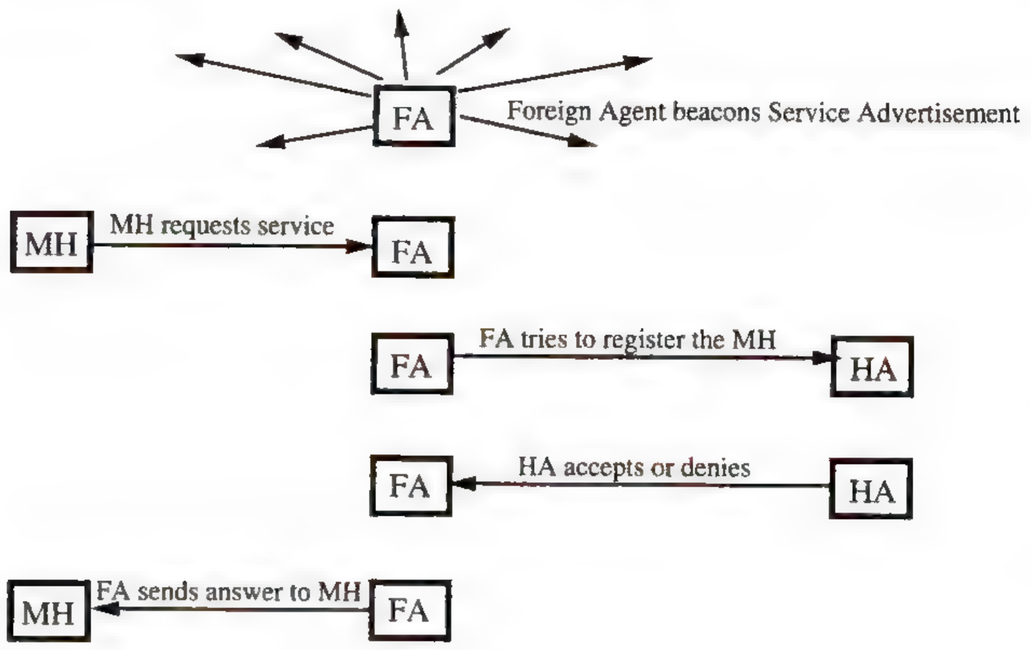


Figure 2: A typical sequence of events for registering a Mobile Host

Remote Redirect

So far, all of the protocol provisions have had the effect of making sure that the Home Agent is able to deliver packets to the Mobile Host. This will satisfy a minimal requirement to get packets delivered to the Mobile Host, because packets addressed to the Mobile Host will find their way to the Home Agent, and from there will be encapsulated and delivered to the appropriate Foreign Agent. However, as discussed previously, requiring the Home Agent to be involved in the delivery of every packet represents a substantial routing inefficiency in many cases. This inefficiency, known as "Triangle Routing" (see Figure 3), can be eliminated by notifying the source of packets destined to the Mobile Host about the current whereabouts of the Mobile Host; in other words, the Correspondent Host could receive and act upon what has been called a "remote redirect." In the figure, the link between the Home Agent and the Foreign Agent is shown as a bold line to indicate that the packet is encapsulated before delivery.

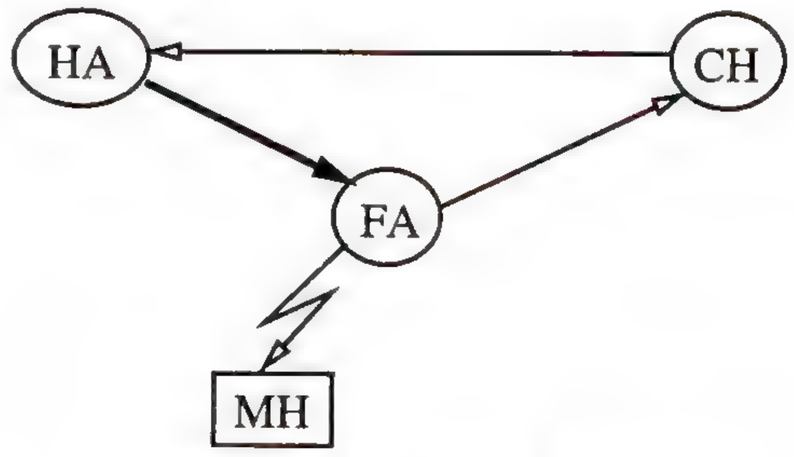


Figure 3: Triangle Routing

Remote redirect is different than the usual redirect, because it specifies that the Correspondent Host should readdress the packet to the Foreign Address of the Mobile Host, instead of the Mobile Host itself. This style of redirection usually specifies a Foreign Address not present on the local network.

Allowing such a feature introduces a new requirement for security, too. When no such redirects occur, the requirement for weak security is automatically met just because all packets to the Mobile Host appear as if they were sent from the Home Agent. And, any Correspondent Host would automatically send packets to the Home Agent, so that no interloper outside the normal path of packets to the Home Agent would be able to intercept these packets.

However, with remote redirect, just as with Loose Source Routing, any interloper computer could cause a correctly functioning Correspondent Host to mistakenly route packets through the interloper instead of through the correct Foreign Agent. Thus, such an interloper could pretend to be any particular Mobile Host, unless additional measures are taken to restore weak security.

The additional measures provide that any Correspondent Host may choose to disregard remote redirects until they are validated by the Home Agent for the target Mobile Host. The Home Agent does indeed have the authentic information, and if the interloper is on the path between the Correspondent Host and the Home Agent, then weak security cannot protect the Correspondent Host in any event. Thus, the Correspondent Host will be as assured as it would otherwise be, that it has gotten good routing information about Mobile Host. This procedure is illustrated in Figure 4.

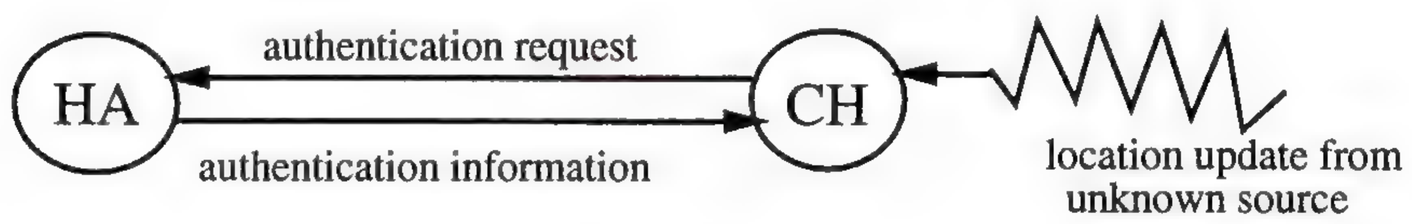


Figure 4: Validating an unsolicited Location Update

The Correspondent Host does not have to have the IP address of the Home Agent to receive this validation; it can just address the validation request directly to the Mobile Host. The Home Agent is specified to intercept validation requests which are addressed to the Mobile Host. These packets will be delivered to the Home Agent anyway, just as any packet to a Mobile Host would be in the absence of remote redirects. The Home Agent then answers the request by consulting its current list of associations between Mobile Hosts and Foreign Agents.

The remote redirect protocol for eliminating Triangle Routing also provides a method for a Mobile Host to notify its previous Foreign Agent of its new Foreign Address whenever the Mobile Host moves. The Mobile Host can send the same remote redirect message to that Foreign Agent. In cases where the Mobile Host does not want to transmit information about its current whereabouts (say, for privacy reasons), it would just notify the previous Foreign Agent that it had moved, without giving a new Foreign Address. When the new Foreign Address is made available, however, the previous Foreign Agent could forward packets to the new Foreign Address instead of causing the packets to be routed back through the Home Agent. In this situation, the magic cookie provided by the Foreign Agent to its clients (the Mobile Hosts) during registration time would be useful to prevent unauthorized remote redirects from effectively disconnecting an unwilling Mobile Host from its current Foreign Agent. The use of such a mechanism can make the forwarding process faster since some Foreign Agents will be satisfied that no additional authentication is necessary.

No help for "ignorant" Hosts

At this time, there is no specified means for allowing unmodified hosts to avoid Triangle Routing. There are ideas, which have not achieved consensus, about how to allow additional cache agents to be placed in strategic locations. These agents could help ignorant hosts by re-addressing packets according to cached location information gleaned from snooping into passing packets.

continued on next page

Mobile IP as seen by the IETF *(continued)*

Layer 2 interface

There are three possible pieces of information that we would like to receive from the protocol layers below the Network Layer (IP in our case). In each case, the availability of this information may offer substantial performance benefits to the overall operation. We expect to provide correct operation even where no additional information is available, but to obtain the necessary information at the Network Layer may incur a performance penalty that would be wholly unnecessary if the information were already available via other means, say through an interface to lower level protocols.

The first useful piece of information would be to discover when a Mobile Host has entered a new cell, or correspondingly when a Mobile Host has exited the effective range of a previous cell. If this information is not available otherwise, the Network Layer in the Foreign Agent might be required to issue periodic beacons to recover the same information. We would like to avoid the need for sending extraneous beacons.

Secondly, it might be very useful at times to be able to discover when there is a momentary loss of the communications channel between the Mobile Host and its Foreign Agent (or, in alternate terms, between the Mobile Station and the Base Station). Armed with this information, we might in the future be able to do some special buffering until the channel is reconstituted. For instance, in the case of an Infrared data channel, the channel might be momentarily interrupted as someone walks behind an obstacle, and then very soon the channel would be usable again.

Lastly, in some situations we could make good use of the MAC address of the currently serving Base Station. This address would not always be useful, but it might be in many circumstances. For instance, if a cell association event occurs but the MAC address does not change, then in many systems the network layer software might determine that there was no need to perform extraneous registrations between the Home Agent and Foreign Agent, or Foreign Agent and Mobile Host.

It is our express intention to design all of our network layer protocols to be independent of the characteristics of lower level protocols, thus none of the suggested information is required. It is only requested for those cases where it makes sense. The working group does not in any way intend to place burdensome requirements upon the protocol specifications of the IEEE 802.11 committee, which is currently undertaking the huge task of specifying a Layer 2 protocol for wireless data communications devices such as radio-frequency and infrared transceivers.

Summary

In summary, the IETF has proposed a basic model of operation which naturally provides the services needed to enable network access for Mobile Hosts. The Mobile Hosts can move freely from one network to another, and their applications will continue to operate just as they would from any existing computer, because the packets to and from the Mobile Host are routed invisibly to the application. It would be substantially more difficult to provide the same level of service between different networks by making modifications at the MAC layer or Logical Link layer of the protocol suite.

We expect that the network operation can be made more efficient if certain parameters and conditions are made known from Layer 2 whenever they are available.

Although we have presented the solution in the framework of providing network access for Mobile Hosts using TCP/IP, there is little within this protocol which is really dependent upon IP, and nothing which depends on TCP. Thus, we believe that the basic solution can be adapted for use with other network-layer protocols. And, there is nothing in the protocol which is machine or operating system dependent.

As this article was being prepared, there has been a small group of people that has been working to get a draft proposal out for comment by mid-October. That draft proposal incorporates most of the design elements discussed within this article. The draft document has been made available for public comment. My hope is that eventually it will have been implemented by enough different people that it achieves the status of Internet RFC, and finally becomes a standard.

Acknowledgment

Although it is inappropriate to single out individuals for particular mention in a group effort such as the IETF, I would like to emphasize that the work presented in this article is largely the product of the whole group, and any individual contributions have benefited greatly from interactions with the rest of the group. I must also emphasize that, while I attempt to fairly represent the ideas and considerations of the IETF Working Group, I do not speak with any special authority. I am not an official representative of the IETF, nor even the Mobile-IP Working Group, even though I have been chosen to provide a liaison between the Working Group and the IEEE 802.11 committee. Other members of the Working Group may well interpret the technical content of past conversations differently than I have here.

References

[1] J. Postel, "Internet Protocol," RFC 791, September 1981.

[2] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[3] J. Postel, "Multi-LAN Address Resolution," RFC 925, October 1984.

[4] J. Postel, "User Datagram Protocol," RFC 768, August 1980.

[5] J. Postel, "Internet Control Message Protocol," RFC 792, September 1981.

[6] R. Braden & J. Postel, "Requirements for Internet Gateways," RFC 1388, June 1987.

[7] C. Hedrick, "Routing Information Protocol," RFC 1388, June 1988.

[8] F. Teraoka, Y. Yokote, & M. Tokoro, "A Network Architecture Providing Host Migration Transparency," Proceedings of ACM SIGCOMM, 1991.

[9] F. Teraoka, "VIP: IP Extensions for Host Migration Transparency," Internet Draft—work in progress, July 1992.

[10] F. Teraoka & K. Claffy and M. Tokoro, "Design, Implementation, and Evaluation of Virtual Internet Protocol," Proceedings of the 12th International Conference on Distributed Computing Systems, June 1992.

[11] F. Teraoka & M. Tokoro, "Host Migration Transparency in IP Networks," *Computer Communication Review,* January 1993.

[12] H. Wada, T. Yozawa, T. Ohnishi, & Y. Tanaka, "Mobile Computing Environment Based on Internet Packet Forwarding," Proceedings of Winter USENIX, January 1993.

continued on next page

Mobile IP as seen by the IETF *(continued)*

[13] J. Ioannidis & G. Q. Maguire Jr., "The Design and Implementation of a Mobile Internetworking Architecture," Proceedings of Winter USENIX, January 1993.

[14] J. Ioannidis, D. Duchamp & Gerald Q. Maguire Jr., "IP-based Protocols for Mobile Internetworking," Proceedings of ACM SIGCOMM, 1991.

[15] J. Ioannides, "Protocols for Mobile Internetworking," PhD Thesis, Columbia University, 1993.

[16] S. F. Wu, C. Perkins, & P. Bhagwat, "Distributed Location Directory Data in Mobile Networking," Proceedings of IEEE Parallel and Distributed Systems, 1993.

[17] C. Perkins, "Providing Continuous Network Access to Mobile Hosts Using TCP/IP," *Computer Networks and ISDN Systems,* November 1993.

[18] A. Myles & D. Skellern, "Comparing Four IP Based Mobile Host Protocols," Joint European Networking Conference, *Computer Networks and ISDN Systems,* November 1993.

[19] K. G. Carlberg, "A Routing Architecture That Supports Mobile End Systems," Proceedings of IEEE MILCOM, 1992.

[20] D. E. Comer, *Internetworking with TCP/IP, Volume I,* Prentice Hall, 1991.

[21] Y. Rekhter & C. Perkins, "Loose Source Routing for Mobile Hosts," Internet Draft—work in progress, July 1992.

[22] C. Perkins & Y. Rekhter, "Support for Mobility with Connectionless Network Layer Protocols," Internet Draft—work in progress, November 1992.

[23] A. Myles & C. Perkins, "Mobile IP," Internet Draft—work in progress, August 1993.

[24] D. Johnson, "Transparent Internet Routing for IP Mobile Hosts," Internet Draft—work in progress, July 1993.

[25] J. Ioannidis, D. Duchamp, G. Maguire, & S. Deering, "Protocols for Mobile Internetworking," Internet Draft—work in progress, June 1992.

[26] J. Penners & Y. Rekhter, "Simple Mobile IP," Internet Draft—work in progress, September 1993.

[27] R. Braden, "Requirements for Internet Hosts—Communication Layers," RFC 1122, October 1989.

[28] R. Droms, "Dynamic Host Configuration Protocol," RFC 1531, October 1993.

[29] CDPD Consortium, "Cellular Digital Packet Data Specification," July 1993.

[30] Dayem, R. A., "A Map to Wireless Networking and Mobile Computing," *ConneXions,* Volume 7, No. 9, September 1993.

[31] Crocker, D., "The ROAD to a New IP," *ConneXions,* Volume 6, No. 11, November 1992.

CHARLES PERKINS holds a B.A. and M.E.E. from Rice University, and an M.A. from Columbia University. Since 1984 he has worked for IBM on a variety of projects related to networks, multiprocessors, and mobile computing. He is a member of the Internet Engineering Task Force, USENIX, IEEE, and ACM. He can be reached on the Internet as: `perk@watson.ibm.com`

The Ten Commandments of Domain Name Service

or

Better Living Through Domain Name Service

by Bryan Beecher, University of Michigan

Introduction

The *Domain Name System* (DNS) provides the mapping between domain-style names (i.e., labels separated by dots, like `news.itd.umich.edu`) and IP addresses (e.g., `141.211.164.30`). The DNS is a distributed database, and has been extended in some ways to do more than just serve up addresses and host names (e.g., *Hesiod*).

The DNS is an interesting system since it is very robust. Even in environments where it is very poorly administered or understood, it still probably works well enough that it meets most people's needs.

I've worked with DNS for over five years, and in that time I've made some contributions to the most widely used software that provides the service (BIND, the *Berkeley Internet Name Domain* server), and occasionally written up a paper on some tool I've written. I've seen very few general articles written during this time on DNS itself. There just seems to be a genuine dearth of available written information that helps people understand what the DNS is and how to manage it effectively. A relatively new book from O'Reilly and Associates ("DNS and BIND" by Liu and Albitz) is very helpful, but I have also heard that some people find an entire book too overwhelming or too daunting.

Thus, this article. It's written for the person who administers the DNS at your site, and it's a simple distillation of some of my experiences with taking care of the DNS over the past few years. The format of the article is a set of ten commandments about the DNS, and without getting too biblical, let's just say that if you follow these commandments, then you'll probably lead a happier life. If you don't, then the DNS will still probably limp along well enough at your site. But your life will be unfulfilled. And now, the commandments:

Vendor software

1. *Thou shalt not run thy vendor's DNS software:* If your DNS is not provided by a UNIX machine, it's permissible to skip this commandment. Otherwise, it is very much in your best interest to pick up the latest copy of BIND. At the time of this writing the latest release is 4.9.2, and it is available from a number of sites across the Internet. You can use *archie* to find out where it is, but one place you can find it for sure is at the University of Michigan on Internet host `terminator.rs.itd.umich.edu` in `/dns/bind4.9.2.tar.Z`).

To be fair, many vendors do try to provide a decent version of DNS software with their products. But even with the best of intentions and circumstances, the vendor's DNS offering just isn't going to be as free of bugs or as full of features as the latest BIND. Running BIND also gives you access to the source code if you're the type to jump right in and find out why something doesn't seem to be working as you think it should, and it gives you a very large audience to consult if you'd rather turn the problem over to the "experts."

Log files

2. *Thou shalt check thy log files for errors, for many server errors are reported there:* Many easy-to-fix errors are reported by *named* when it parses the cache file and zone files during an initial load or a reload. Better still, *named* usually provides you with the file name and the line number where the error occurred.

continued on next page

The Ten Commandments of DNS *(continued)*

Some errors are simple typos, while others are semantic in nature (e.g., a name has both a CNAME record and other record types associated with it). Tailing the log after an edit and reload is a good idea, as well as getting daily e-mail extracts of errors reported in the logs.

Aliases

3. *If thou hast a nameserver listed by the root servers, than thou ought to use an alias like* ***dns.<your-domain>*** *to refer to it:* Many sites use a single machine for a lot of their needs, e-mail, DNS, news, and so on. If this nameserver happens to serve a domain that is delegated by the root servers (e.g., `UMICH.EDU, 211.141.IN-ADDR.ARPA`) and also happens to provide some of these other services, especially e-mail, it is a really good idea to use some kind of alias for the machine when listing it in the root servers. It can take a couple of days under the best of circumstances to have the InterNIC update the root servers if you change the IP address of one of these servers, and so there ends up being a timing problem since you'd like to have both the root servers and your own servers list the same data.

For example, we use the name `DNS.ITD.UMICH.EDU` as an alias for all of our domain nameservers at U-M that serves the `UMICH.EDU` domain. We list a single name at the InterNIC, and that single name maps to as many IP addresses as there are machines performing service at any given moment in time. We use the alias name `DNS2.ITD.UMICH.EDU` for our one machine that does secondary service. We don't use either of these names for e-mail, and so even if we add a domain nameserver, or move one to another network, it won't interfere with mail.

CNAME, MX, and A

4. *Thou shalt know when to use a* ***CNAME****, when to use an* ***MX****, and when to use an* ***A*** *record:* Many times there is a need for a single machine to be known by more than one name. Sometimes there is a need for a machine that no longer exists to continue to be listed in the DNS, just because it is so well known across the Internet. When should you use a CNAME record? Or an A record? Or just an MX record?

- *CNAME:* want name to be transparent for things like mail, won't conflict with other records, and doesn't have multiple CNAMEs (e.g., `news.itd.umich.edu`).

We quite often use CNAME records for machines that have moved from one domain to another (and so all of its records are listed in one domain, and the other domain has just a CNAME pointing to the other name), and for IP-based services that we may want to move from one machine to another at some point in time. In this latter case, a CNAME is more attractive to us than an A record since then we can accept mail addressed to `<service>.<domain>` without changing the mailer configuration.

- *MX:* for mail only, would like the name to disappear from mailing lists one day (e.g., `umix.cc.umich.edu`).

We use MX records in two main situations. One is where we have many desktop machines in a domain, but we want mail addressed to those machines to go to a central mail hub, usually a larger machine that is well run, has backups done on it, and so on. In this case, each one of these desktop machines gets a single MX record pointing to the hub, and the hub's mailer configuration is changed to accept mail for not only the hub itself, but all of the desktop machines with the MX records.

Another situation in which we use MX records is for a host that was very well known at one point across the Internet, and thus there are many entries on mailing lists containing that host name. In this case we use an MX record since the only service we wish to continue is e-mail; if someone tries to use *ftp* or *telnet* or *finger* to that host, it will fail. If that person then follows up to our postmaster, they can be pointed in the right direction. In the interim, we continue to handle mail for that name.

• *A:* want all services to work, or want multiple records mapped to the same name (e.g., `dns.itd.umich.edu`).

The biggest way we use A records—besides simply listing IP addresses for machines—is when we want to be able to send mail to a name that is also a domain. For example, the User Services branch of our Info Tech Division uses a name like `us.itd.umich.edu`, and machines are placed under that domain. But, we also want to be able to handle e-mail sent to `<user>@us.itd.umich.edu`. In this case a CNAME won't work at all since this name is going to have other records associated with it (and a major DNS faux pas is to list a name with both a CNAME record and other records), which leaves using either an A record or MX record. And since we often want services like *telnet* to work along with e-mail, we use the A record.

Log files again

5. *Thou shalt check thine log files a second time for expired domains and failed zone refreshes, for it is better to find such things sooner than later:* If your domain nameserver does not perform any secondary service, you can ignore this commandment. If it does perform such service, it is important to check the logs each day. There are a few problems that can show up.

One, sites can start listing you as a nameserver without asking first. This will show up as a message like: `"forw: query(<name>) contains our address (<server-name>:<server-IP-address>)"`. In this case, you should either go ahead and start up secondary service for the site, or call them up and ask them to delete the offending NS record pointing to your nameserver. Either way, you should let them know that they should ask *before* listing you.

Two, sites will make mistakes with their nameserver, and then not notice. For example, syntax errors in the boot file or zone file can cause a nameserver to drop a domain. This will show up as a message like: `"named-xfer: bad response to SOA query from [<master-server>], zone <zone-name>: rcode nn, aa nn, ancount nn, aucount nn"`.

Three, sites will start using a new scheme with their SOA serial numbers. A classic example is that they go from a date-based scheme (e.g., 940215) to a simple one (e.g., 1). This will show up as a message like `"Zone <zone-name> SOA serial# (<serial-number>) rcvd from [<master-server>] lower than ours (<serial-number>)"` in the log files, and will mess up zone refreshes until they either return to their original scheme, or until you perform a zone refresh by hand.

Four, zones can expire if errors are not corrected soon. This will show up simply as `"secondary zone '<zone>' expired"` in the log files. In this case call up the hostmaster of that zone, and let them know they have a problem.

continued on next page

The Ten Commandments of DNS *(continued)*

Delegation

6. *Thou shalt choose a scalable way of delegating domains:* It is very important to think big with the DNS even if your site is currently small. Once a location within your site has been given a domain name, they will want to keep it. Forever. (This commandment is from personal experience having failed to do this.)

Using a naming scheme of `<machine>.FOO.EDU` or `<machine>.BIG.COM` is soon going to break down. It leaves no room for delegating domains to other DNS admins, and it is also going to lead to very large zone files which will likely be harder to manage.

Four-part names of the form `<machine>.<dept>.<site>.<top-level>` seem to be the ones most commonly used. I think this level of separation works well enough for most organizations.

Very large sites can often benefit from using an extra layer in the name. While some parts of the site may not like the longer names, it is really the only way to ensure that the DNS stays manageable in the long run. (There are even those that argue that the DNS won't be enough, and that ISO X.500-style naming will replace the DNS. Usually mentioning this, and giving an example of such a name is enough to make the person grateful for the five-part DNS name.)

At the U-M we use four- and five-part names. The five part names are used with very large parts of the University like the liberal arts college and the medical center `<machine>.<department>.lsa.umich.edu` and `<machine>.<dept>.med.umich.edu`, respectively. This works well for me since I can then turn over an entire college or school to a DNS admin, and then it can be up to him or her to add new sub-domains as more and more departments attach to the campus network and need DNS. And since the big colleges and departments often have a central computing organization, this has worked out really well. Unfortunately, many administrative departments are also going on-line and do not have a collective IT provider internally, nor do they have any kind of common root in the DNS tree other than `UMICH.EDU` , and these become the bulk of the four-part names like `<machine>.ro.umich.edu` for the Registrar's Office and `<machine>.admin.umich.edu` for very high level administrators who don't have their own domain (e.g., the President of the University).

SOA records

7. *Thou shalt not use random values in your SOA records:* The *Start of Authority* (SOA) record is one of the most important, and least understood, records used in the DNS.

There should be exactly one SOA record in any zone file. We like to place ours at the top of the file, followed by the nameservers for the zone, and then followed by all of the other records in the zone. The SOA record lists seven pieces of information, most of which are very important.

First the SOA record lists the name of the machine on which the zone file is maintained. This is going to be a primary server for this zone. Currently, this field isn't used for a whole lot, and so its accuracy is not critical. But there isn't any reason *not* to keep it up-to-date.

Second, the SOA lists the e-mail address of the contact for the zone. Since an @-sign is not a legal character according to the DNS specs, people often substitute a dot (.) for the @-sign like this: `bryan.umich.edu` instead of `bryan@umich.edu`.

Many people also like to use a "hostmaster" alias for routing DNS mail. If you use such an alias in your SOA record, be sure that you have actually listed such an alias in your mailer's alias file! Also be sure that whatever address you list actually works. I don't know how many times I've found a problem with someone's DNS, went to mail them, and then had the mail bounce.

Next, the SOA record lists five numbers:

The first of these is the zone's *serial number*. It doesn't matter what you use as long as the number always goes up when the file is modified. Some people like to use simple integers, but we've found that using a number that represents the date seems to work well too, and it also gives people some idea of the last modification date of the zone file. So, we use a format like *yyyymmdd* or just *yymmdd,* like 19930207 or 930207. Some people also like to add an extra digit in case they change the zone file more than once in a single day; personally I just "borrow" from tomorrow and advance the serial number into the future if need be. Decimal points are allowed in serial numbers, but are treated differently that you'd probably expect. There is no good reason to use decimal points in serial numbers, and they will just burn you some day. Stay away from them!

The second number is the *refresh time.* This is the period in seconds that any secondary servers should wait between attempting zone refreshes, the process that secondary servers use to detect and retrieve new data from master servers. A secondary server thinks its data is old if its copy of the SOA serial number is less than the one it gets from the master server. If it has old data, it asks the master server to send it a complete copy of the zone file, not just the modified records. Values from one hour to one day are appropriate depending upon how often you update your data and how long you are willing to wait for your secondary server to be updated. We use eight hours, and if there is ever a case where our secondary server just *has* to be updated immediately, I give the hostmaster over there a call on the phone and ask him to do a zone transfer by hand.

The third number is the *retry time.* If the secondary server fails to contact the master server at the refresh time, then the secondary server should retry using this number as the period instead of the refresh time. We use a smaller number here, four hours.

The fourth number is the *expiration time.* This is also used by the secondary server only. This specifies how long the secondary server should continue to serve the domain if it continues to fail to perform a zone refresh. This number can be a little tricky to choose. On one hand, a small number is nice so that you can identify problems earlier rather than later. On the other hand, a large number is also nice so that secondary servers will continue to work even if your server is down for an extended period of time. Since we follow Commandment 2, and check our log files each day, we opt for the latter thinking, and use a value of one month for this.

The last number is the *default time-to-live (ttl) value* used on records within the zone file, or the minimum time-to-live value, or both! Some versions of the DNS software used this number as the default ttl for records in the zone file (the ttl is the amount of time that non-authoritative nameservers—that is, not the master server, and not any secondary servers—will cache a DNS record before purging it), and they also used it as the minimum value allowed. BIND 4.9.2 uses it only as the default, not the minimum value.

continued on next page

The Ten Commandments of DNS *(continued)*

Here you want to select a value which is good for both your nameserver and the network (a high number) but one which also prevents nameservers from keeping potentially old data around for a long time (a low number). We use four hours, which is probably a little on the low side. A value of twelve or 24 hours is probably a better idea.

The cache file

8. *Thine cache file shalt contain only root servers:* An important file in the configuration of any nameserver is the cache file, or the cache preload. This contains the bootstrapping information that the nameserver needs in order to work. This file is sometimes called `named.ca`, or `root.cache`, or `named.cache`. The name doesn't matter, but the contents does. This file should contain a list of the root servers, and only the list of the root servers. A copy of this file is available via anonymous FTP from `internic.net` in `/domain/named.cache`.

This file isn't modified very frequently (last modification at the time of this writing was April 21, 1993). However, it is important that you do keep your version of it up-to-date. If you don't, and it contains the names and addresses of domain nameservers that are no longer acting as root servers, then you will cause your own nameserver to perform poorly, and even worse, your erroneous data may creep into other people's nameservers.

Down the tree

9. *Thou shalt keep thy secondaries well informed:* One of the lamest things you can do with the DNS is create something called a "lame delegation." A lame delegation is when there is an NS record listing a nameserver as authoritative for a domain, but that nameserver isn't actually serving that domain. This usually happens in one of two ways (see also the next commandment).

One way is that people create new domains, list NS records in the zone file for that domain, and never bother to tell the site that is performing the secondary DNS. This happens all the time with sites that have a class B address, and initially ask some site to perform secondary service for `xxx.nnn.IN-ADDR.ARPA` and a handful of domains under that domain corresponding to the handful of class C subnets of `nnn.xxx.0.0` that are currently in use. Then, inevitably, the site will begin to hand out more subnets, create more domains, and list the secondary site's nameserver in NS records. But they forget to let the secondary site know! And so that nameserver doesn't actually perform secondary service for these sites until someone notices by happenstance.

Up the tree

10. *Honor thy parents:* Besides creating a lame delegation by forgetting to notify your secondary servers, you can also create a lame delegation by changing a nameserver, and forgetting to tell your parent domain. Quite often this seems to happen to sites when they upgrade machines, and move their nameserver from one IP address to another. The site will then contact the InterNIC, and let them know that the nameserver listed in the root servers for `FOO.EDU` has changed, but they forget to tell the InterNIC about their `IN-ADDR.ARPA` domains.

A very simple rule is to remember that whenever you change some NS records, or the A record that goes along with a name listed in an NS record, you must let the hostmaster at the domain one level higher know too. That way your records will match those of the domain's parent domain. This is very important to do since the DNS search algorithm always begins at the root of the tree, and works its way down to the answer.

And so if a parent domain is listing old information, then the search may end up going to the wrong place, which wastes bandwidth, and makes for longer DNS reply times.

Running the DNS isn't really all that difficult. Making it run well so that little problems don't turn into big problems, and planning for the future takes a little more work. It's this latter aspect that doesn't seem to be well documented in the available literature. I hope this article helps you with your part of the DNS, and I'll know that I really did a good job if someone says something like "So, *that's* what that thing is for!"

References

[1] P. Mockapetris, "Introducing Domains," *ConneXions,* Volume 1, No. 6, October 1987.

[2] J. B. Postel & J. K. Reynolds, "Domain requirements," RFC 920, October 1984

[3] C. Partridge, "Mail routing and the domain system," RFC 974, January 1986.

[4] M. K. Stahl, "Domain administrators guide," RFC 1032, November 1987.

[5] M. Lottor, "Domain administrators operations guide," RFC 1033, November 1987.

[6] P. V. Mockapetris, "Domain names—concepts and facilities," RFC 1034, November 1987.

[7] P. V. Mockapetris, "Domain names—implementation and specification," RFC 1035, November 1987.

[8] B. Manning, "DNS NSAP RRs," RFC 1348, July 1992.

[9] "The Directory—overview of concepts, models and services," CCITT X.500 Series Recommendations, December 1988.

[10] S. E. Kille, "Replication and distributed operations extensions to provide an internet directory using X.500," RFC 1276, November 1991.

[11] S. E. Kille, *Implementing X.400 and X.500: The PP and QUIPU Systems,* Artech House, 1991, ISBN 0-89006-564-0.

[12] M. T. Rose, "Realizing the White Pages using the OSI Directory Service," Technical Report 90–05–10–1, Performance Systems International, Inc., May 1990.

[13] S. Benford, "Components of OSI: X.500 Directory Services," *ConneXions,* Volume 3, No. 6, June 1989.

[14] Marshall T. Rose, *The Little Black Book: Mail Bonding with OSI Directory Services,* Prentice Hall, Inc., ISBN 0-13-683210-5, 1992.

[15] *ConneXions,* Special Issue on Electronic Mail and Directory Service, Volume 6, No. 9, September 1992.

BRYAN BEECHER works for the Information Technology Division of the University of Michigan. Since 1988 he has made several contributions to the Berkeley Internet Name Domain server (BIND) including a set of tools which identifies DNS "lame delegations" and notifies the appropriate parties. Currently he is a leader in the effort to move the U-M from a mainframe-based computing environment to one based on distributed computing and client–server relationships. Bryan has an A.B. and M.S. from the University of Michigan. E-mail: `bryan@umich.edu`

Book Review

Ithiel de Sola Pool, *Technologies without Boundaries: On Telecommunications in a Global Age,* Harvard Press, 1990, 283 pages.

Introduction

Ithiel de Sola Pool was Chairman of the Political Science department at MIT, advisor to high officials, and a noted political scientist. It is as an interpreter of the effects of the information revolution that he is best known. When Pool died in 1984, he left behind a manuscript meant to be his *magnum opus.*

This manuscript was picked up by Eli Noam, also a noted political scientist, and finally surfaced as *Technologies without Boundaries.* Why a book that had been in the works for several years in 1984 should still be relevant in 1990 (or 1994) in a field that changes as quickly as a politician's positions is a question that immediately comes to mind.

The reason *Technologies without Boundaries,* like its predecessor *Technologies of Freedom,* is still relevant is due to the penetrating analysis and insight of its author. We can see that even years later his grasp of the social, political, and economic effects of technology are right on the mark.

Technology and policy

Pool's fundamental contribution is to demonstrate conclusively that technology does not determine policy. He shows that many assumptions about computer technology and networks, such as the truism that computers centralize power in the hands of a few, are false. Pool proves to be a powerful voice for individuals and liberty, seeing an opportunity in computers and communications for increased liberty, not decreased freedom.

Pool takes an extremely wide, technically accurate view of communications. He has an intimate grasp of history, from the printing presses of Korea in the 13th century to the intricate details of Vail's creation of the AT&T empire.

Trends

He shows how each revolution, from talking to writing to printing to electronic communication, has had fundamental impacts on how society structures itself. Our latest revolution, still underway has already had profound effects. Distance has ceased to be a barrier; different media such as speech, text, and pictures have merged; computing and communications have combined to allow us to manipulate information instead of just transmit it; individuals have access to individualized technology.

At first, the digital revolution meant increasingly wide broadcasts. The power press extended the reach of newspapers into the thousands, the rotary press extended it even further to tens of thousands by the mid 1800s, and by the mid 1900s newspapers like *Pravda* were reaching millions and broadcasts by networks reached tens of millions.

That trend has been reversing. Small magazines, bulletin board systems, electronic mail, video recorders, cheap broadcasting equipment, all allow increased diversity. While it used to be that there were only a few radio stations, they are now commonplace. Individuals can now easily establish a printing press with a Commodore and BBS software.

Diversity has been accompanied by changes in spatial patterns of interaction. National and geographic limitations are becoming less relevant. People are organizing themselves into virtual communities.

These changes have not come easily, nor are they complete. Governments try and protect their role through mercantilist policies. Pool analyzes restrictions on transnational data flows through national policies and international groups like the CCITT.

Pool's analysis of international trade concludes that protectionism will not work in the long run. He sees resources shifting into data havens and prohibitions on transnational data flows as ultimately futile. Pool is no blind free market advocate, however. He is sensitive to questions of erosion of national sovereignty, protection of cultural values, and even the preservation of government prerogatives.

Criticism

Some of his most piercing criticism is reserved for the CCITT and the PTTs behind it. As with the previous book, *Technologies of Freedom,* he is not kind to policy makers that refuse to recognize the reality of technology. He is adamant that policy must take into account technology: policy is not determined by technology and hard choices must be made with a grip on reality.

Valuable contribution

Technology without Boundaries is a thoughtful, well-researched analysis of communications technology. At times, the book rambles a bit, particularly when trying to catalog different properties of technology (e.g., copper wire versus optical fiber). For the most part, however, Pool balances the requirements to inform the technical novice with an extremely sophisticated analysis of the political, economic, social, and ecological implications of computer networks. *Technology without Boundaries* lacks the polish of *Technologies of Freedom,* but is still a valuable contribution. (See sidebar "Some Related Books").

—Carl Malamud

Some Related Books

Ithiel de Sola Pool, *Technologies of Freedom,* Harvard University Press, 1983. A brilliant analysis of the fundamental values of the U. S. Constitution as applied in the press, the broadcast media, the telephone, and computer-based communications.

David H. Brandin and Michael A. Harrison, *The Technology War: A Case for Competitiveness,* Wiley-Interscience, 1987. An examination of competitiveness in international trade as applied to information technology.

Simon Nora and Alain Minc, *The Computerization of Society: A Report to the President of France.* MIT Press, 1980). A classic study that identified IBM as one of the biggest threats to French national sovereignty.

Letters to the Editor

Dear *ConneXions,*

In my review of Raman Khanna's book *Distributed Computing* I made one of those minor but egregious errors that requires correcting. I specifically cited the highly practical chapter on "Migration Strategies"; it's the type of chapter that is valuable by virtue of the scars it lets the reader see. My error was in citing only Bob Morgan as its author, although Roland Schemers was a co-contributor. I would appreciate your publishing my apology for the error.

Thanks,
—Dave Crocker, Silicon Graphics

[Ed.: The following letter was originally received via fax, although we have reason to believe that it originated as an electronic message]:

Here in Cyberspace, we've been observing Jon Crowcroft's journey toward virtual reality with much interest. The trail of burnt matches and faxes makes it easy to follow. However, we lost him for a little while at Universal Cybervisity London (UCL) until he turned up in the Cyberion City shuttle, escaping from a Moebius loop in the MBONE routing system.

You may rest assured that he is alive (?) and well here in Cyberspace, though seemingly fixated on burning faxes and chanting mantras about something he calls Afterspace.

—Virtual Vint

p.s. if you warm this fax lightly, a secret message will appear...

OOPS, WATCH OUT! Now you've done it! .....CTL-ALT-DEL..

Announcement and Call for Papers

The *1994 USENIX Symposium on High-Speed Networking* will be held August 1–2, 1994 at the Claremont Hotel & Resort in Oakland, California.

Goals

The goals of this symposium are to encourage the UNIX and high-speed networking communities to commingle, to examine the issues and trends in high-speed networking, and to explore the impact of high-speed networks on systems and applications design. High-speed, high-capacity networks promise to change the way we compute, much as faster processors and large memories have done in the past. Fast, wide-area networks pose fresh challenges even for mature operating systems, such as UNIX. How will these innovations shape the design of future operating systems? Can we devise applications that fully (and productively) consume the bandwidth at our disposal?

Schedule

This single-track symposium offers two days of technical presentations, followed by a field trip to two high-speed networking testbeds, XUNET/BLANCA and CalREN, in Berkeley on the third day. Formally reviewed papers will be presented and published in the Symposium Proceedings. A copy of the Proceedings will be distributed to all attendees; additional copies may be purchased from the USENIX Association.

Topics

We seek presentations of original, previously unpublished work on these (and related) topics:

- Network architectures
- Operating system support for high-speed networks
- Protocols
- Performance
- Network management
- Applications
- Practical experiences

Submission guidelines

If you are interested in presenting your work at the symposium, please submit an extended abstract as described below. The extended abstract should represent the final paper in "short form." Its object is to persuade the Program Committee that you will deliver a good 20–25 minute presentation and final paper. The Committee needs to know that authors:

- Are tackling a significant problem.
- Are familiar with the current literature about the problem.
- Have devised an original solution.
- Have implemented the solution and, if appropriate, have characterized its performance.
- Have drawn appropriate conclusions about what they have learned and why it is important.

Note that the Program Committee considers it unethical to submit the same paper simultaneously to more than one conference or publication, or to submit a paper that has been or will be published elsewhere, without disclosing this information with the submission.

If your paper is accepted, you are expected to provide a full paper in camera-ready form for publication in the Proceedings and to present your work at the Symposium.

A typical extended abstract is roughly 2500 words (5 pages). Indicate clearly whether the paper represents a design, an implementation or a system that is in wide use. You are encouraged to include references. Supporting material may be in note or outline form. If you wish, you may supplement the extended abstract with a copy of a full paper. Please submit one copy of an extended abstract using at least *two* of the following methods: E-mail (preferred method) to: **net94papers@usenix.org**; Mail to: Pat Parseghian, Program Chair, (see address below); Fax to: Pat Parseghian +1 (908) 582-5857.

Please, with your submission, include the following information about the author(s):

- Name (indicate which author will serve as the contact)
- Affiliation
- Daytime telephone
- Postal address
- E-mail address
- FAX number

For more program information

Refer questions about refereed paper submissions and other program concerns to the Program Chair:

Pat Parseghian
AT&T Bell Laboratories
Room 2C–472
600 Mountain Avenue
PO Box 636
Murray Hill
New Jersey 07974–0636
USA
Fax: +1 (908) 582-5857.
Telephone: +1 (908) 582-4229
E-mail: **pep@research.att.com**

Registration information

Materials containing full details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be available early June 1994. If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert Street
Suite 613
Lake Forest
California 92630
USA
Telephone: +1 (714) 588-8649
Fax: +1 (714) 588-9706
E-mail: **conference@usenix.org**

Important dates

Extended abstracts due:	May 2, 1994.
Notification to authors:	May 16, 1994.
Camera-ready final papers due:	June 20, 1994.
Registration materials available:	June 1994.

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen

Printed on recycled paper